

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

5           The petitioners, Carrel W. EWING, a citizen of the United  
States and a resident of Incline Village, Nevada, and whose post  
office address is 285 Deer Court, Incline Village, NV 89451;  
Brian P. AUCLAIR, a citizen of the United States and a resident  
10 Falls Loop, Reno, NV 89511; Mark J. BIGLER, a citizen of the  
United States and a resident of Eugene, Oregon, and whose post  
office address is 3422 Willamette Street, Eugene, OR 97405;  
Andrew J. CLEVELAND, a citizen of the United States and a  
resident of Reno, Nevada, and whose post office address is 5419  
15 Greenvview Court, Reno, NV 89502; James P. MASKALY, a citizen of  
the United States and a resident of Sparks, Nevada, and whose  
post office address is 1568 Topeka Circle, Sparks, NV 89434;  
and, Jay Henley WILLIAMS, a citizen of the United States and a  
resident of Yankton, South Dakota, and whose post office address  
20 is 803 West Fourth Street, Yankton, SD 57078; pray that letters  
patent may be granted to them for a

POWER-MANAGER CONFIGURATION UPLOAD AND DOWNLOAD  
METHOD AND SYSTEM FOR NETWORK MANAGERS

25           set forth in the specification.

FILED "09/03/00"

POWER-MANAGER CONFIGURATION UPLOAD AND DOWNLOAD  
METHOD AND SYSTEM FOR NETWORK MANAGERS

5

CO-PENDING APPLICATIONS

This Application is a continuation-in-part of United States  
Patent Application serial number 09/732,557, filed 12/08/2000,  
titled NETWORK-CONNECTED POWER MANAGER FOR REBOOTING REMOTE  
10 COMPUTER-BASED APPLIANCES, that is a continuation-in-part of  
United States Patent Application serial number 09/375,471, filed  
08/16/1999, titled REMOTE POWER CONTROL SYSTEM THAT VERIFIES  
WHICH DEVICES IS SHUT-DOWN BEFORE SUCH ACTION IS COMMITTED TO,  
which in turn is a continuation-in-part of United States Patent  
15 Application serial number 08/685,436, that was filed on  
7/23/1996 and is titled, SYSTEM FOR READING THE STATUS AND  
CONTROLLING THE POWER SUPPLIES OF APPLIANCES CONNECTED TO  
COMPUTER NETWORKS, and now United States Patent 5,949,974,  
issued 09/07/1999.

20

BACKGROUND OF THE INVENTION

Field of the Invention

25

The invention relates generally to remote power  
management systems, and more particularly to uploading and  
downloading user configuration information in bulk to avoid  
time-consuming and tedious power-manager configuration  
dialogs and scripts.

30

Description of the Prior Art

Remotely located power managers at network equipment  
centers typically need to be configured by either the  
35 installer or network administrator. When only a few power  
controllers and options are being handled, such configuration  
management is next to trivial. But large network server

farms, hotels, and other concentrated-equipment sites are being put in all around the world. The numbers of controlled equipment, and the numbers of sites can be overwhelming. Manually peeking and poking at the configuration from a network administrator's control panel is not very feasible for these major installations.

#### SUMMARY OF THE PRESENT INVENTION

Briefly, a power management system embodiment of the present invention comprises a power manager with a network agent in communication over a network with a network manager. The power manager is connected to control the operating power flowing to various pieces of computer network equipment at a single site. A user is able to assign names to each control port, and the power manager maintains a list of enrolled users who have access. Many operational mode choices are possible, and each can be configured by the user while remote from the power manager. The power manager can be commanded to upload a user configuration list, and it can be commanded to accept a downloaded user configuration list.

An advantage of the present invention is that a system and method are provided that can speed the configuration of a power manager.

Another advantage of the present invention is that a system and method are provided for duplicating, modifying, and replicating user configurations for power managers.

A further advantage of the present invention is that a system and method are provided that allows a network console operator to control the electrical power status of a router or other network device.

A still further advantage of the present invention is that a system and method are provided for reducing the need

for enterprise network operators to dispatch third party maintenance vendors to remote equipment rooms and POP locations simply to power-cycle failed network appliances.

Another advantage of the present invention is that a system and method are provided for reducing the time it takes to restore a failed network appliance and improving service levels.

Another advantage of the present invention is that a system and method are provided for reducing organization losses from network downtime.

These and many other objects and advantages of the present invention will no doubt become obvious to those of ordinary skill in the art after having read the following detailed description of the preferred embodiments which are illustrated in the various drawing figures.

#### IN THE DRAWINGS

Fig. 1 is a functional block diagram of a power manager system embodiment of the present invention that is able to pass user configuration data over a computer data network like the Internet;

Fig. 2 is a functional block diagram of a power manager embodiment of the present invention showing the configuration and flow of user configuration files and display screens;

Fig. 3 is a functional block diagram of a number of power managers in a system embodiment of the present invention that have been connected in a daisy-chain serial arrangement to expand the number of power control points possible;

Fig. 4 is a plan view of a layout of a printed circuit board for a power manager in one embodiment of the present invention that has reduced EMI levels; and

Figs 5A and 5B are functional block diagrams of how a power manager with only four logic-level power-control points and modular output connectors can be expanded to serially control sixteen power-control points.

5

#### DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

10 Fig. 1 represents a power manager system embodiment of the present invention, and is referred to herein by the general reference numeral 100. A network management system (NMS) 102 is connected by a network 104 to a remote site 106. A power controller 108 forwards operating power through a  
15 sensor 110 and relay-switch 112 to a computer-based appliance 114. Such operating power can be the traditional 110 VAC or 220 VAC power familiar to consumers, or direct current (DC) battery power familiar to telephone central-office "plant" employees. A network interface controller (NIC) 116 may be  
20 used to connect the computer-based appliance 114 to the network 104. Such would be especially true if the computer-based appliance 114 were a server, router, bridge, etc.

The power controller 108 can be configured to operate in a number of different modes, and such options are selected  
25 and stored in a configuration memory. The NMS 102 may download configurations to power controller 108, and may upload them for editing, archiving, and/or duplication to other power controllers 108 at other remote sites 106. Embodiments of the present invention are directed towards  
30 systems and methods that do such uploading, downloading, editing, archiving, and duplication of power manager configuration files.

The power manager system 100 maintains the operating health of the computer-based appliance 114. Such computer-  
35 based appliance 114 is prone to freezing or crashing where it is effectively dead and unresponsive. It is also some

09892350-062601

5  
10

15  
20

25

30

35

display 130 and keyboard 128 can also be virtual, in the sense that they are both emulated by a Telnet connection over the network 104.

The NMS 102 typically comprises a network interface  
5 controller (NIC) 132 connected to a computer platform and its  
operating system 134. Such operating system can include  
Microsoft WINDOWS-NT, or any other similar commercial  
product. Such preferably supports or includes a Telnet  
10 application 136, a network browser 138, and/or an SNMP  
application 140 with an appropriate MIB 142. A terminal  
emulation program or user terminal 144 is provided so a user  
can manage the system 100 from a single console.

If the computer-based appliance 114 is a conventional  
15 piece of network equipment, e.g., as supplied by Cisco  
Systems (San Jose, CA), there will usually be a great deal of  
pre-existing SNMP management software already installed,  
e.g., in NMS 102 and especially in the form of SNMP 140. In  
such case it is usually preferable to communicate with the  
network agent 122 using SNMP protocols and procedures.  
20 Alternatively, the Telnet application 136 can be used to  
control the remote site 106.

An ordinary browser application 138 can be implemented  
with MSN Explorer, Microsoft Internet Explorer, or Netscape  
NAVIGATOR or COMMUNICATOR. The network agent 122 preferably  
25 includes the ability to send http-messages to the NMS 102 in  
datapackets 126. In essence, the network agent 122 would  
include an embedded website that exists at the IP-address of  
the remote site 106. An exemplary embodiment of a similar  
technology is represented by the MASTERSWITCH-PLUS marketed  
30 by American Power Conversion (West Kingston, RI).

Many commercial network devices provide a contact or  
logic-level input port that can be usurped for the "tickle"  
signal. Cisco Systems routers, for example, provide an input  
that can be supported in software to issue the necessary  
35 message and identifier to the system administrator. A device  
interrupt has been described here because it demands

100-000000-000000

Network information is generally exchanged with protocol data unit (PDU) messages, which are objects that contain variables and have both titles and values. SNMP uses five types of PDUs to monitor a network. Two deal with reading terminal data, two deal with setting terminal data, and one, the trap, is used for monitoring network events such as terminal start-ups or shut-downs. When a user wants to see if a terminal is attached to the network, for example, SNMP is used to send out a read PDU to that terminal. If the terminal is attached, a user receives back a PDU with a value "yes, the terminal is attached". If the terminal was shut off, a user would receive a packet informing them of the shutdown with a trap PDU.

Once a user has installed and configured the power manager 108, a serial communications connection is established. For example, with a terminal or terminal emulation program. Commercial embodiments of the present invention that have been constructed use a variety of communications access methods.

For direct RS-232C access, a user preferably starts any serial communication software that supports ANSI or VT100



5

10

20

25

35

currently operating with into a file. The user at user terminal 144 is also able to send a command to the power manager 124 to have it accept a power manager configuration file download. The download file then follows. Once  
5 downloaded, the power manager 124 begins operating with that configuration if there were no transfer or format errors detected. These commands to upload and download configuration files are preferably implemented as an extension to an already existing repertoire of commands, and  
10 behind some preexisting password protection mechanism. HyperTerminal, and other terminal emulation programs allow users to send and receive files.

In a minimal implementation, the power manager configuration files are not directly editable because they  
15 are in a concentrated format. It would, however be possible to implement specialized disassemblers, editors, and assemblers to manipulate these files off-line.

Fig. 2 represents a power management system 200 that includes a power controller 202. A microprocessor 204 is  
20 connected to a non-volatile memory 206 that can store user configuration data, e.g., user-assigned names for various controlled ports. A plurality of power control relays and sensors 208 respond to various control signals issued by the microprocessor 204 and are dependent on the user  
25 configuration data. A serial input/output (I/O) port 210 allows the configuration data to be inspected or modified via a data terminal 212. For example, a SHOW display screen 214 on the terminal 212 graphically displays all user configuration choices in the memory 206.

30 Alternatively, the user configuration can be transferred in a file 216 over a data modem pair 217-218 to a remote terminal 220. The file 216 may also be uploaded or downloaded from a disk 222. An off-line editor 224 allows the user configuration file to be created, uploaded-modified,  
35 and downloaded. For example, where many power management systems 202 are being installed or maintained, the user

T09250\*0925850

configuration file 216 corresponding to each can be individually customized by the editor 224. Such could save a great deal of time, effort, and expense.

Fig. 3 is a diagram of an expandable power management system 300. In one commercial embodiment of the present invention, a first power controller board 302 is daisy-chain connected through a serial cable 303 to a second power controller board 304. In turn, the second power controller board 304 is connected through a serial cable 305 to a third power controller board 306. All three power controller boards can communicate with a user terminal 308 connected by a cable 309, but such communication must pass through the top power controller board 302 first.

Alternatively, the user terminal can be replaced by an IP-address interface that will provide a webpresence and interactive webpages. If then connected to the Internet, ordinary browsers can be used to upload and download user configurations.

Each power controller board is preferably identical in its hardware and software construction, and yet the one placed at the top of the serial daisy-daisy-chain is able to detect that situation and take on a unique role as gateway. Each power controller board is similar to power controller 202 (Fig. 2). Each power controller board communicates with the others to coordinate actions. Each power controller board independently stores user configuration data for each of its power control ports. A typical implementation will have four relay-operated power control ports. Part of the user configuration can include a user-assigned name for each control port.

A resynchronization program is executed in each microprocessor of each power controller board 302, 304, and 306, that detects where in the order of the daisy-chain that the particular power controller board is preferably located. The appropriate main program control loop is selected from a collection of firmware programs that are copied to every

power controller board. In such way, power controller boards may be freely added, replaced, or removed, and the resulting group will resynchronize itself with whatever is present.

The top power controller board 302 uniquely handles interactive user log-in, user-name tables, its private port names, and transfer acknowledgements from the other power controller boards. All the other power controller boards concern themselves only with their private resources, e.g., port names.

During a user configuration file upload, power controller board 302 begins a complete message for all the power controller boards in the string with the user-table. Such is followed by the first outlets configuration block from power controller board 302, and the other outlet configuration blocks from power controller boards 304 and 306. The power controller board 302 tells each when to chime in. Each block carries a checksum so transmission errors can be detected. Each block begins with a header that identifies the source or destination, then the data, then the checksum.

During a user configuration file download, power controller board 302 receives a command from a user that says a configuration file is next. The user-name table and the serial-name table is received by power controller board 302 along with its private outlets configuration block and checksum. The next section is steered to power controller board 304 and it receives its outlets configuration block and checksum. If good, an acknowledgement is sent to the top power controller board 302. The power controller boards further down the string do the same until the whole download has been received. If all power controller boards returned an acknowledgement, the power controller board 302 acknowledges the whole download. Operation then commences with the configuration. Otherwise a fault is generated and the old configuration is retained.

Fig. 4 is a plan view of a power controller printed circuit board (PCB) 400, in an implementation of power

TOP SECRET

controller boards 302, 304, and 306. A microprocessor (CPU) 402 comprises a Dallas Semiconductor DS80C320, which is a CMOS derivative of an Intel 8051. A pair of programmable array logic (PAL) devices 404 and 406 do I/O port and memory address decoding. A socket 408 is provided for a two M-bit flash memory device that will ultimately store the user configuration data. A PIC-microcontroller 410, e.g., a Microchip Technology (Chandler, AZ) PIC16F84A device, talks with a temperature sensor, controls serial communication data rates, and secures the flash memory in socket 408. It further is responsible for LED blink synchronization amongst the power controller boards. Four serial I/O connectors 411-414 are RS-232 interfaced by a set of level conversion interface chips 416-419. A three-terminal regulator 420 provides operating power for the digital logic. A pair of analog matrix devices 422 and 424, e.g., CD74HC22106 types, allow the serial I/O ports to be interconnected in a variety of arrangements. A set of four RJ-11 type connectors 426-429 respectively connect to four power-control relay boards (see Fig. 5) that actually switch high-current utility circuits to network appliances. A set of two-wire connectors 430-433 can be used to forewarn each network appliance that its utility power is about to be interrupted, e.g., so a graceful shutdown can be executed.

Such PIC-microcontroller 410 preferably includes a code protection bit that can be permanently set to make its internal program code unreadable. Such then allows the PIC-microcontroller 410 to act as a tamper-protection device for the whole PCB 400, and especially for the program code and data executing and passing through CPU 402 from the flash memory.

Electromagnetic interference (EMI) is now a major concern among governments and equipment manufacturers/users. So the PCB 400 has been implemented as a four-layer board with its signal layers on the outside and the powerplanes on

the inside. Such construction has been found to reduce EMI to acceptable levels.

DB-9-connectors had been used on previous products, but it was learned that IDC-10 type connectors were more reliable and did not express the micro-shortening behavior of the conventional connectors. These IDC-10 connectors allow mass termination with ribbon conductor cable types, and so are easier to manufacture.

Fig. 5A represents a basic power control system 500 that includes four single-point relay boards 501-504 that are able to independently control the operating power flowing to various pieces of network equipment and other appliances. Each relay board 501-504 is separately connected to a power manager 506, e.g., with a three-wire cable 507-510 and RJ-11 type plugs and jacks. A user can control the system 500 from a user terminal 512.

Fig. 5B represents an expanded power control system 520 that includes four four-point relay boards 521-524. This array is able to independently control the operating power flowing to sixteen pieces of network equipment and other appliances. Each relay board 521-524 is separately connected via a serial RS-232 communications link to a power manager 526, e.g., with a three-wire cable 527-530 and RJ-11 type plugs and jacks. A user can control the system 520 from a user terminal 532. Preferably, the power managers 506 and 526 differ only in their programming, and not in their constituent hardware. Logic level relay boards require only two-wires (control signal and common), but serial relay boards require three wires (data send, data receive, and common). Even logic level boards use three wires, with the third wire being used for the relay board to report power output status (on or off) back to the power controller circuit board.

Each relay board 521-524 includes a PIC-microcontroller, e.g., a Microchip Technology (Chandler, AZ) PIC16F84A device, that controls the serial communication interface with the

power manager 526. Serial data is interpreted by the microcontroller and is used to independently operate each of the relay board's several onboard relays. Such serial communication and therefore the microcontroller isn't  
5 necessary for the relay boards 501-504 (Fig. 5A).

In a preferred application, the expanded power control system 520 is used instead of daisy-chain connecting power managers to get more control points. For example, power controller boards 304 and 306 (Fig. 3) could be eliminated  
10 and still as many as sixteen control points can be accommodated. The configuration in Fig. 3 would otherwise accommodate twelve control points as shown.

In general, power controller embodiments of the present invention will have a number of configuration parameters that  
15 are set by users. These include descriptions of the type of power controller hardware present, e.g., the type, number and capabilities of the various intelligent power modules (IPM's) and sensors. The power controller configuration attaches names to power control relays (PCR's), and other components.  
20 The configuration can be used to switch serial connections, and control usernames, passwords and access to control of power controller PCR's based on the login username.

The power controller configuration data is preferably located in a single 256-byte memory on each board. Even in a  
25 daisy-chain of power controller boards, each memory is used to store information about the board it is on. The memory at least contains all configuration information specific to the particular power controller board. A first board in a power controller board daisy-chain further includes a list of  
30 defined users and the passwords associated with the users, and a list of alias names associated with serial connections that can be accessed via a power-controller CONNECT command.

The first board in a power controller board daisy-chain uniquely maintains a username and password table for user  
35 login validation. User access to particular PCR's is based on an access bit map for each PCR associated with the power

TOP SECRET

controller board. When the first board in a daisy-chain has verified a username, a user identification number associated with the username is activated and is used by all boards to allow access to various power controller resources. Saving and restoring power controller configuration information involves the 256-byte memory on each board in the daisy-chain. The username and serial name tables for all boards in a daisy-chain are kept only on the first board in the power controller board.

10       A command retrieves configuration data from the power controller, and another sends the data to the power controller. Initially, the configuration data is passed between users and the power controller via an ASCII file upload and download. Such upload and download is performed via a standard terminal emulation program such as the Windows HYPERTERM program. Once the code is completed on the power controller to allow this simple mechanism to save and restore power controller configuration, the next phase of this development effort is to create a Windows based program that will automate the configuration upload and download. In addition to automating the upload/download procedure for Windows users, this program further provides a graphical user interface program to allow Windows users the ability to view and manipulate a power controller configuration file.

25       A power controller configuration save/restore command includes various parameters. Depending on the action requested by the user (via the command parameters) the command interacts with the user to either set up an ASCII file upload or download. The ASCII data that is sent or received contains the power controller configuration information.

35       An SCFG command is used initiate a configuration upload or download, depending on a parameter supplied, e.g., SCFG SAVE|RESTORE. The SCFG command is used to copy the power controller configuration information from the power controller when the SAVE parameter is specified, and to copy

TO SED-052680



the power controller configuration information to the power controller when the RESTORE parameter is specified. If the command is simply used with no parameters, an error message is displayed that lists the allowable parameters. If the

5 SAVE parameter is specified, the user is prompted to press "Q" for quit, "C" for continue, or "D" for delay. The continue option will cause the power controller to start sending configuration data immediately. The delay option inserts a sixty second delay before sending the configuration

10 data.

When the power controller has sent all of its configuration data, a message is displayed, "Configuration Send Complete - stop receiving ASCII data". The usual power-controller prompt is displayed signaling the upload is

15 complete.

If the RESTORE parameter was specified, a message, "Ready to receive ASCII configuration data - begin sending ASCII configuration data" is displayed. The power controller begins parsing any received data for configuration

20 information. The configuration data download is aborted if no data is received for sixty seconds, or if a Ctrl-C is received. If the data received does not conform to the expected configuration data format, the download is rejected. An appropriate message is displayed.

The power controller preferably waits to display any message until after the last character is received. Such delay helps ensure a configuration data stream does not obscure the message. The power controller automatically restarts using the existing configuration if the download was

25 aborted or rejected. The configuration data transmissions preferably include checksums to improve data integrity. If the data and checksums add up wrong, the power controller rejects the data as being corrupted.

30

As long as the configuration data is not detected as

35 being corrupted, the power controller will use as much of the download configuration data as possible to update the current

configuration. Too much data for the board combination is not considered to be corrupted data. For example, it is possible to download configuration data for a ten-board power controller board daisy-chain to a two-board power controller board daisy-chain, or to download configuration data for a two-board power controller board daisy-chain to a ten-board power controller board daisy-chain. The power controller configuration does not revert to defaults if no configuration data is received, the current configuration remains in effect if no configuration data is received.

Once the power controller has successfully received an entire configuration data file, a message is displayed, "Configuration data successfully received for N power controller boards....This power controller board daisy-chain contains M power controller boards....WARNING - Only Partial Configuration Data Received....Press Y)es to use this data or N)o to discard this data".

The WARNING message is only displayed if the number of power controller boards that the configuration file refers to does not match the number of power controller boards actually in the power controller board daisy-chain. The user then responds with a "Y" to cause the power controller to restart using the data or with an "N" to cause the power controller to restart using the previous configuration. The "Y" and "N" responses can be entered in either upper or lower case. If the users does not respond to the message within the standard five minute power controller prompt time out, the power controller will restart using the previous configuration data.

The power controller will display, "Restarting using configuration" or "Restarting using existing configuration". The SCFG command requires administrative username privileges to execute.

The configuration data that is sent to or from the power controller is sent in seven-bit ASCII format. There are two general data configuration blocks that are contained in the

power controller configuration data. One of the general configuration blocks is used for comments while the other contains the actual power controller configuration data. During normal data stream parsing of a power controller configuration data stream, the processing code is sensitive to four possible events.

If sixty seconds elapse between characters in a power controller configuration data stream, the parsing assumes a problem exists with the configuration data and the process is aborted with appropriate messages. None of the data in a power controller configuration data stream is used. Such time out detection is active during all power controller configuration data stream parsing.

If a CTRL-C character is detected in the power controller configuration data stream, the parsing assumes the process is being aborted by user intervention and the process is aborted with appropriate messages. None of the data in a power controller configuration data stream is used. Such abort character detection and processing is active during all power controller configuration data stream parsing.

A comment data configuration block indicator (e.g., the "#") is received, if a "#" character is detected in the power controller configuration data stream, processing switches to the comment block data stream processing. Comment character parsing is only active during normal data stream parsing when neither a comment nor a configuration data block is being parsed.

A configuration data block indicator (e.g., the "\$") is received, if a "\$" character is detected in the power controller configuration data stream. Processing switches to the configuration block data stream processing. Configuration data block character parsing is only active during normal data stream parsing when neither a comment nor a configuration data block is being parsed.

A power controller configuration comment block is recognized by a "#" character. When the configuration data

stream is being parsed in the normal data stream parsing mode, a "#" indicates the beginning of a general comment block. When a "#" character is detected, normal data stream processing switches to comment block data stream processing.

5 During comment block data stream processing, all data in the configuration data stream after the "#" character is treated as comment data until a carriage return or line feed character is received. The sixty-second time out and the CTRL C parsing and actions continue in effect.

10       The carriage return character (0x0D) signifies the end of a comment block and causes the parsing of a power controller configuration data stream to return to the normal data stream processing as described earlier.

The line feed character (0x0A) signifies the end of a comment block and causes the parsing of a power controller configuration data stream to return to the normal data stream processing.

There are no checksums associated with comment blocks and no additional checks for data integrity are performed on comment blocks. Once a carriage return or line feed is detected in the comment block data, normal data stream processing continues. It is possible for garbage characters to exist in the configuration data stream between comment blocks and configuration data blocks. Garbage characters that occur between valid configuration blocks are ignored (unless the character is a CTRL C which aborts the process).

A power controller configuration data record block is recognized by a "\$" character. When the configuration data stream is being parsed in the normal data stream parsing mode, a "\$" indicates the beginning of a configuration data record block. When a "\$" character is detected, normal data stream processing switches to data record block data stream processing. During data record block data stream processing, the data that follows the "\$" character is expected to be in the following format:

bytes 1 and two  
bytes 3 and 4  
byte 5 and 6  
bytes 7 ... 516  
5 last two bytes  
count field  
board ID field  
record type field  
record data field  
10 checksum field

The first two characters after the "\$" represent the high nibble and the low nibble of a one-byte hexadecimal value in the range of 00 through FF, e.g., 0-255 decimal.

15 Because the two characters in the count field represent a single hexadecimal byte, the characters are limited to the ASCII-0-9 digits and the upper case ASCII characters A through F. The value of the hexadecimal byte represented by the two characters in the count field is the actual number of

20 hexadecimal bytes that are represented by the characters in the record data field. The count is not the actual number of characters that are transmitted in the record data field data stream since the representation of a single hexadecimal byte requires two characters (one character for each nibble). The

25 maximum size of a power controller configuration data record block is five hundred eighteen characters. Two characters for count, plus two characters for board-ID plus two characters for record type, plus up to five hundred ten characters for record data, and two characters for checksum.

30 The two characters after the count field represent a one-byte hexadecimal value in the range of 00-1A (e.g., 0-26 decimal) that is the power controller board ID that the record data is associated with. The "00" value is a special value that is used to indicate the associated data record

35 applies to all power controller boards in a daisy-chain. Other values are used to identify a specific power controller

1099990056604

The two characters after the board ID field represent a one-byte hexadecimal value in the range of 00 through 04 that is the record type identifier for this data record block. Currently five record types are defined as follows:

- 10

20

25

35

stream will result in a value of zero. The checksum applies to actual hexadecimal values represented by the characters in the data stream, not to characters in the data stream. All of the fields in the configuration data record block (e.g., count field, board ID field, record type field, and record data field) are included in the sum to which the checksum value is applied.

The configuration data general header/version record is the identified by an ASCII-01 in the record type field. Such record provides information on the entire contents of the configuration data stream. Such record must be the first configuration data record block in a configuration data stream. Configuration data comment blocks may precede the current record in the configuration data stream, but the first configuration data record block must be a header/version record.

The count field of the current record is set to ASCII-04 to reflect the number of bytes of actual data that are contained in the record data field. The board ID field of the current record is set to ASCII-01 to indicate that the current record is directed towards the first board in the daisy-chain. The record data field of the general header/version record is formatted as follows:

bytes 1 and two
bytes 3 and 4
bytes 5 and 6
bytes 7 and 8
version
number of username table entry records
number of serial port table entry records
number of power controller board page configuration table records

The first two characters in the record data field are an ASCII representation of the current version of this

20

35



high and low nibbles of the actual hexadecimal bytes that make an entry in the power controller username table. The actual number of ASCII characters in the record data field is eighty (e.g., two characters for each hexadecimal byte). As with all configuration data record blocks, the last two characters contain the checksum value.

The serial port table entry record is identified by an ASCII-03 in the record type field. Such record is used to send an entry in the power controller serial port table.

These records can appear in any order in the configuration data stream except as the first or last configuration data record block in a configuration data stream. The count field of the current record is set to ASCII twenty-eight (e.g., forty decimal) to reflect the number of bytes of actual data that are contained in the record data field. The board ID field of the current record is set to ASCII-01 to indicate that the current record is directed towards the first board in the daisy-chain. The record data field of the serial port table entry record contains ASCII characters that represent the high and low nibbles of the actual hexadecimal bytes that make an entry in the power controller serial port table. The actual number of ASCII characters in the record data field is eighty, e.g., two characters for each hexadecimal byte. As with all configuration data record blocks, the last two characters contain the checksum value.

The power controller board page configuration table entry is the identified by an ASCII-04 in the record type field. Such record is used to send an entire 256-byte power controller board configuration data area. The single byte count field in the configuration data record block limits the actual amount of data that can be represented in a record to two hundred fifty-five bytes. As a result, the first byte of the power controller board configuration data area is not included in the data. Such byte is used internally by the power controller code for virtual power controller purposes and is not required nor desired to be included in the

configuration data. These records can appear in any order in the configuration data stream except as the first or last configuration data record block in a configuration data stream.

5       The count field of the current record is set to ASCII FF  
to reflect the number of bytes of actual data that are  
contained in the record data field. The board ID field of  
the current record is set to reflect the ASCII representation  
of the unit ID of the power controller board to which the  
10      record applies. The record data field contains ASCII  
characters that represent the high and low nibbles of the  
actual hexadecimal bytes of the power controller board  
configuration data area. The actual number of ASCII  
characters in the record data field is five hundred ten,  
15      e.g., two characters for each of the two hundred fifty-five  
hexadecimal bytes. As with all configuration data record  
blocks, the last two characters contain the checksum value.

The configuration data general trailer record is the identified by an ASCII-00 in the record type field. Such record is used to signal the end of the configuration data stream. Such record must be the last configuration data record block in a configuration data stream. Configuration data comment blocks may precede or follow the current record in the configuration data stream, but no additional configuration data record blocks may follow the current record. The count field of the current record is set to ASCII-00 to indicate there is no data in the record data field for the current record. The board ID field of the current record is set to ASCII-01 to indicate that the current record is directed towards the first board in the daisy-chain. As with all configuration data record blocks, the last two characters contain the checksum value.

An SCFG command initiates a configuration upload/download. The first board in a power controller board daisy-chain assumes control of all upload/download operations. A command line processor on the first board

5

10

15

20

35

processor on the first board in the daisy-chain sends an internal power controller command to all of the boards in the power controller board daisy-chain that informs them that a configuration data stream is to be received. Such is similar to a SET CODE UPLOAD command that puts all of the boards into a receive data mode, and causes a current configuration to be saved to flash ROM. However this command does not return to the system monitor. The power controller code that processes this command prepares to receive the configuration data and sends a positive response to the first board in the daisy-chain to indicate the configuration upload state is active. The first board in the daisy-chain waits for a response from all boards in the daisy-chain and then displays the download message to prompt the user to begin the configuration data send. A timer is set to allow the user five minutes to begin the configuration download. If the download fails for time out, invalid data detected, etc., an error message is displayed for twenty seconds. Then a power controller restart is initiated. The error message that is displayed in the event of an error is only displayed after a ten second time period has elapsed since the last character was received from the user. Such delay is incorporated to prevent the error message from colliding with an incoming data stream and therefore being lost.

The parsing that occurs during the data transfer is described in the power controller configuration data format section of this document. Individual power controller boards process data that is directed to them as necessary. If errors are detected, the individual power controller boards record the error and wait for the end of command condition to report the error.

Once the configuration data general trailer record is received, all boards verify their received data. Such verification includes verifying version number agreement between the configuration data and the existing data as well as checksum verification. The first-in-daisy-chain board

TO 990 "0902550

propagates DTR down the daisy-chain and waits for a response from each board. Boards in the daisy-chain send a "+" to indicate the configuration was received and is verified. The boards send a "-" to indicate the received data was not verified. The boards send a "?" to indicate no configuration data was received for this board. If any boards report invalid data, the entire operation is aborted and the power controller is reset using the previous configuration data.

5 The first-in-daisy-chain board evaluates the responses from the other boards in the daisy-chain and sends a message to the user indicating the result of the command and prompting the user to commit the configuration. If the number of boards in the power controller board daisy-chain does not correspond to the number of configuration data records received or if some of the boards responded with a "10", a warning message is displayed in addition to the normal prompt messages to the user. Such warning message is to alert the user to discrepancy between the configuration data and the actual power controller board daisy-chain being 15 configured. 20

Based on the users response to the prompt, the first-in-daisy-chain board sends a power controller internal command to the daisy-chain to signal the boards to save or ignore the configuration. The first board then displays the success message and after a twenty second delay, initiates a power controller restart. At this point the power controller will restart with either the configuration or with the existing configuration depending on the users response. 25

Although the present invention has been described in terms of the present embodiment, it is to be understood that 30 the disclosure is not to be interpreted as limiting. Various alterations and modifications will no doubt become apparent to those skilled in the art after having read the above disclosure. Accordingly, it is intended that the appended claims be interpreted as covering all alterations and 35

TO BE REPRODUCED

modifications as fall within the true spirit and scope of the invention.

What is claimed is:

0989230 062604